AMENDMENT TO THE CLAIMS

Please **AMEND** claim 20 as follows.

Please **ADD** claim 21 as follows.

A copy of all pending claims and a status of the claims are provided below.

1. (Original) A method of reordering data packets received out of order, the method comprising the steps of:

reading context information from a received data packet to determine whether the received packet is in a given sequence;

comparing said context information of the received data packet to an expected sequence count for the given sequence, and storing the received packet with said context information in a memory as a linked list when there is a match, all received packets in the linked list being in order;

creating a new linked list each time a new data packet is received out-of-order;

linking in order all subsequent packets received in order to the new linked list;

constructing a reorder table of addresses of the first packet for all linked lists; and

reading packets out of the memory in an order specified by the reorder table.

2. (Original) The method of claim 1, wherein the comparing step includes incrementing the expected sequence count.

3. (Original) The method of claim 1, wherein comparing context information includes comparing a flow type indicator.

4. (Original) The method of claim 1, further comprising the step of constructing a transmission table of one or more entries, each entry including at least one of a flow indicator, a sequence number and a memory address associated any of the linked lists, the flow indicator being associated with the reorder table.

5. (Previously Presented) The method of claim 4, wherein:

the constructing a reorder table includes one or more entries, each entry having a sequence number of the first packet of one linked list;

the reading packets step includes the steps of: accessing the one or more transmission table entries in turn, and for each entry, using the flow indicator to locate the associated reorder table;

searching the associated reorder table to locate any entry having the sequence number matching the accessed transmission table entry's sequence number; and

reading all packets in order from a linked list associated with the located any entry.

6. (Original) The method of claim 5, wherein the constructing a reorder table step includes constructing one or more reorder tables, each of the one or more reorder table associated with one or more packet flows.

7. (Original) The method of claim 5, wherein the accessing the one or more transmission table entries, in turn, provides a relative ordering among all the read data packets between one or more packet flows.

8. (Previously Presented) A method for ordering packets, the method comprising the steps of:

detecting at least one of an in-sequence and an out-of-sequence packet chain in one or more packet flows;

storing the detected at least one of the in-sequence and the out-of-sequence packet chain in a memory;

providing a sequence number with each of the stored in-sequence and the out-of-sequence packet chain;

associating the sequence number with an address in the memory of at least one of the stored in-sequence and the out-of-sequence packet chain;

ordering the at least one of the in-sequence and the out-of-sequence packet chain from the memory based on the associated sequence number to provide one or more packet flows all in-sequence; and

creating a linked list each time a new data packet is received out-of-sequence and linking in order all subsequent packets received in sequence to the linked list.

9. (Previously Presented) The method of claim 8, wherein the detecting step includes the steps of:

initializing an expected sequence count for each of the one or more packet flows;

comparing a received sequence count to the expected sequence count for the packet flow associated with a currently received packet;

setting the expected sequence count for the packet flow associated with the currently received packet to the received sequence count when unequal; and

incrementing the expected sequence count for the packet flow associated with the currently received packet.

10. (Original) The method of claim 8, wherein the storing step includes linking one or more received packets into a linked list associated with the sequence number of each of the at least one in-sequence and the out-of-sequence packet chain.

11. (Original) The method of claim 8, wherein the sequence number is a list of sequence numbers, each associated with at least one of the in-sequence and the out-of-sequence packet chain.

12. (Original) The method of claim 8, wherein the associating step includes the steps of:

determining if a context switch is necessary by checking a packet context information in a received packet; and

switching context when the packet context information has changed for a next received packet, the packet context information including flow context information.

13. (Original) The method of claim 8, wherein the associating step further includes entering an entry so that a temporal order of the at least one of the in-sequence and the out-of-sequence packet chain is maintained.

14. (Original) The method of claim 8, further including building a table by entering the sequence number of a first packet of any of the packet chains and an address in the memory of the stored at least one of the in-sequence and the out-of-sequence packet chain of the any of the packet chains.

15. (Original) The method of claim 8, wherein the ordering step includes the steps of:

accessing a transmission table to retrieve a next non-null entry;

searching a reorder table based on the next non-null transmission table entry to locate a reorder table entry with a lowest sequence number and at least equal to or less than the sequence number in the next non-null transmission table entry;

transmitting all packets in the at least one of the in-sequence and the out-of-sequence packet chain identified in the located reorder table entry so that the packets are sent in sequence order, and incrementing a transmitted packet count for each transmitted packet; and removing the located reorder entry.

16. (Original) The method of claim 15, further including the step of returning to the transmission table to access another next non-null entry.

17. (Original) The method of claim 15, wherein the reorder table and the transmitted packet count are associated with one of the one or more packet flows.

18. (Original) The method of claim 15, wherein the searching a reorder step includes searching the reorder table based on the next non-null transmission table entry to locate the reorder table entry with the sequence number next to be sent as indicated by the transmitted packet count.

19. (Original) The method of claim 8, wherein the ordering step includes:

retrieving an entry from a transmission table associated with one or more reorder tables;

identifying locations of the in-sequence and out-of-sequence packet chains;

searching the associated one or more reorder tables for the sequence number associated with one of the in-sequence and out-of-sequence corresponding to the retrieved entry;

determining whether the sequence number associated with the one of in-sequence and out-of-sequence packet chain is a minimal sequence number;

if so, then transmit in order the packet chain associated with the minimal sequence number; and if not the minimal sequence number, then search the reorder table for next minimal sequence number and transmit in order the packet chain associated with next minimal sequence number and continue search and transmitting next minimal sequence number and associated packet chain until the retrieved minimal sequence number and associated packet chain has been transmitted.

20. (Currently Amended) A computer program product comprising a computer usable medium having readable program code embodied in the medium to perform a method operable to:

detect at least one of an in-sequence and an out-of-sequence packet chain in one or more packet flows;

store the detected at least one of the in-sequence and the out-of-sequence packet chain in a memory;

provide a sequence number with each of the stored in-sequence and the out-of-sequence packet chain;

associate the sequence number with an address in the memory of at least one of the stored in-sequence and the out-of-sequence packet chain; ~~and~~

order the at least one of the in-sequence and the out-of-sequence packet chain from the memory based on the associated sequence number to provide one or more packet flows all in-sequence; and

create a linked list each time a new data packet of the packet chain is received out-of-sequence and linking in order all subsequent packets received in sequence to the linked list.

21. (New) The method of claim 1, wherein all subsequent packets are linked based on a sent order.